D-skyline and T-skyline Methods for Similarity Search Query in Streaming Environment

Ling Wang[1], Tie Hua Zhou[1], Kyung Ah Kim[2], Eun Jong Cha[2], and Keun Ho Ryu[1]*

[1]Database/Bioinformatics Laboratory, School of Electrical & Computer Engineering,
Chungbuk National University, Chungbuk, Korea
{smile2867, thzhou, khryu}@dblab.chungbuk.ac.kr
[2]Department of Biomedical Engineering, Chungbuk National University, Chungbuk, Korea
{kimka, ejcha}@chungbuk.ac.kr

Abstract. There has been a concerted effort in recent years to build data stream management systems for a specific streaming application. Requirement for the lowest space usage and fast response, the traditional skyline is not suit for streaming data process. Two approaches are proposed to solve this problem, namely D-skyline and T-skyline. These two methods are more excellent to adapt to this kind of data characters that are huge, vary, distributed, and coming in a high-speed rate. Focus on similarity search in streaming environment; our proposed methods give almost "real" results in an approximate way.

Keywords: skyline, similarity search, stream processing.

1 Introduction

In many stream applications [1, 2, 3], similarity search is more practical than exact match in stream processing, where both query and data are always changed over time. The length of multi-streams could be very large, since new values are continuously appended. Therefore, the similarity of multi-streams is expressed by means of the last values of each stream, using a sliding window approach. The naïve approach is to delete the old items by using timestamp techniques, to re-apply the data reduction mining technique on the new items, and finally store the resulting summary only in the access memory to do the further final approximated result analysis. This process is very efficiently both in CPU time and numeric items calculated by one-pass processing. Since multi-streams are usually too large to be stored in main memory, skyline algorithms for similarity search are used in the sense that emerged as an important summarization technique happens in the main memory. Several algorithms [4, 5] have been proposed targeting the efficient skyline evaluation on large datasets. These solutions always classified into two categories, depending on whether they assume an index. Intuitively, the index-based schemes are faster than index-independent strategies, since they avoid accessing the entire data collection, yet their applicability is significantly limited by the indexing requirement. They may not to be

[*] Corresponding author.

indexed which the data are dynamically produced in many streaming applications (such as moving sensors, predicted analysis). Therefore, the traditional techniques may not suit for streams exactly. Our proposed D-skyline and T-skyline is more excellent to adapt to this kind of data characters.

2 Skyline Operator and Future Work

D-skyline and T-skyline are focus on requirement for the lowest space usage and fast response to the users in a high accuracy guarantee results. D-skyline algorithm organizes already computed multi-streams into sub-windows such that candidate similar search tuples could quickly prune when they are dominated by some other streams. For a candidate query, only their distance under the threshold could be calculated and stored into sub-windows. In a defined time series, only the frequent times for each stream need to be shown in each sub-window, then the most appeared streams as an approximated result of similar search queries for the whole sliding window domain. Other unsigned items would be removed in order to release space for continuously coming new streams. T-skyline is similar to D-skyline except one pass on the sub-windows, which need top-k items as a list only. One thing is that T-skyline may use the lowest space usage than others and give a very exactly answers to satisfy a more fast response. We demonstrate that both our methods are efficiency as data reducing mining techniques for similarity search over multi-streams distributed processing. The important thing is that they give more exactly approximate results as lower as possible on space usage. In the future, we will show a more detailed discussion on the difference between these two methods after evaluation experiments.

Acknowledgments. This work was supported by the Korea Institute of Energy Research (KIER) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0000478).

References

1. Nehme, R.V., Rundensteiner, E.A., Bertino, E.: Tagging Stream Data for Rich Real-Time Services. Journal of VLDB Endowment, Vol. 2, Issue: 1, pp. 73-84 (2009)
2. Vu, T. H. N., Park, N.K., Lee, Y.K., Lee, Y.M., Ryu, K.H.: Online discovery of Heart Rate Variability patterns in mobile healthcare services. Journal of Systems and Software. Vol. 83 Issue: 10, pp. 1930-1940 (2010)
3. Lee, Y.K., Shin, J.P., Kim, K.D., Ryu, K.H.: An adaptive data storage and historical query processing for storage-centric sensor network. Journal of Innovative Computing, Information and Control, Vol.7, No.5, pp. 2945-2959 (2011)
4. Zhang, S., Mamoulis, N., Cheung, D.W.: Scalable Skyline Computation Using Object-based Space Partitioning. In: 35th SIGMOD international conference on Management of data, pp. 483-494. ACM Press, Providence (2009)
5. Zhang, S., Mamoulis, N., Kao, B., Cheung, D.W.L.: Efficient Skyline Evaluation over Partially Ordered Domains. Journal of VLDB Endowment, Vol. 3, Issue: 1, pp. 1255-1266 (2010)